UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/622,047	08/23/2000	Alexandr Andreevich Moldovyan	P65855US0	4150

136 7590 08/17/2007
JACOBSON HOLMAN PLLC
400 SEVENTH STREET N.W.
SUITE 600
WASHINGTON, DC 20004

EXAMINER	
LANIER, BENJAMIN E	
ART UNIT	PAPER NUMBER
2132	
MAIL DATE	DELIVERY MODE
08/17/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

41

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 09/622,047
Filing Date: August 23, 2000
Appellant(s): MOLDOVYAN ET AL.

John C. Holman
Reg. No. 22,769
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 08 May 2007 appealing from the Office action mailed 14 September 2006.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

No amendment after final has been filed.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

Schneier, Bruce, Applied Cryptography, 1996, John Wiley & Sons, Pages 270-273.

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claims 1, 3, 5 are rejected under 35 U.S.C. 102(b) as being anticipated by Schneier. Referring to claim 1, Schneier discloses the DES algorithm wherein a 64-bit block of plain text is split into a right half and a left half (Page 270), which meets the limitation of breaking down a data block into N≥2 subblocks. The encryption key is broken up into 16 subkeys (Figure 12.1), which meets the limitation of generating an encryption key in the form of a set of subkeys. There are 16 rounds of identical operations in which the data are combined with the key (Page 270 & Figure 12.1). The operations performed are exclusive or (Xor) operations (Page 270 & Figure 12.1), which meets the limitation of alternatively converting said data subblocks by performing a two-place operation on the data subblock and the subkey because figure 12.1 shows the exclusive or operation being performed on the subblock and the subkey. An exclusive or (Xor) operation is a two-place operation. In each round the key bits are shifted depending on a subblock (Page 270 & Figure 12.1), which meets the limitation of prior to carrying out said two-place operation on an I-th data subblock and a subkey, an operation of permuting subkey bits is performed on the subkey depending on the value of the j-th data subblock where i is not equal to j, because figure 12.1 shows that the data subblock Lo is exclusive or'd with the result of the permutation function on subkey Ki that depends on data subblock Ro. Therefore, from figure 12.1 (looking at one round for an example), Lo would meet the limitation of the i-th data subblock, Ro would meet the limitation of the j-th data subblock, K1 would meet the limitation of the subkey being permuted, and function f would meet the limitation of the permutation function.

Referring to claim 3, Schneier discloses that each round the subkey bits are shifted depending on a subblock as discussed above (Page 270 & Figure 12.1), which meets the limitation of an operation of an operation of cyclic offsetting subkey bits depending on the j-th subblock is used as the j-th subblock-dependent operation of permuting subkey bits because the permutation function on the subkeys are performed each round and that would be considered cyclical.

Referring to claim 5, Schneier discloses that the subkeys are shifted as a result of a permutation function that depends on the j-th data subblock as discussed above (Page 270 & Figure 12.1). In the later rounds of the algorithm (see figure 12.1, specifically the calculation for R2), the permutation function operates on subkey K2 dependant upon the result of R1 data block calculation. The R1 data block calculation involved subkey K1, and therefore, the calculation of R2 also includes data from subkey K1, which meets the limitation of the operation of permuting subkey bits is performed on one of said set of subkeys depending on the value of the j-th data subblock, where i is not equal to j, and the value of another subkey.

(10) Response to Argument

Appellant argues, "Schneier, when describing algorithm DES (US Data Encryption Standard), does not disclose any feature of converting a subkey depending on data being converted." This argument is not persuasive because Schneier discloses in the "Outline of the Algorithm" section of page 270 (second paragraph):

> The right half of the data is expanded to 48 bits via an expansion permutation, combined with 48 bits of a shifted and permuted key via an XOR, sent through 8 S-boxes producing 32 new bits, and ***permuted again*** (emphasis added).

From this outline (and corresponding Figures 12.1 & 12.2), it is clear that this final permutation function (corresponding to "...and permuted again"), which is shown in figure 12.2 as the "P-Box Permutation", takes as an input the XOR of a data subblock (Ri-1 from Figure 12.2) and a subkey (output of the "Compression Permutation" from Figure 12.2). Therefore, the P-Box Permutation operation effectively permutates the subkey dependant upon the value of a j-th data subblock because the subkey was combined with the data subblock (via XOR) prior to being permutated using the P-Box Permutation.

Appellant characterizes the procedures of Schneier on page 4 of the Appeal Brief and states, "conversion of the broadened subblock by means of its addition with 48-bit subkey X: = X ⊕ K (before this step, no conversions depending on the data block being converted have been performed on this round subkey, i.e. **no permuting subkey bits depending on data has been performed**." Examiner notes that this operation was not relied upon to meet the claim limitations. This operation correponds to the XOR of the output of the "Expansion Permutation" and the "Compression Permutation" from Figure 12.2. While the Examiner has relied upon the "P-Box Permutation" to read on the claimed permutation functions.

Appellant further characterizes the procedures of Schneier on page 4 of the Appeal Brief stating, "performing the transmutation operation ***P*** which consists in a fixed permutation of the vector **Z** bits, i.e. permutation of the vector **Z** bits is performed independently of the value of some data subblock but always in the same manner, as prescribed by the Schneier reference. After performing operation ***P***, the value of *f*(R, K) is obtained, i.e. we have *f*(R, K) = ***P(Z)***." This argument is not persuasive because the issue is not whether "permutation of the vector **Z** bits is performed independently of the value of some data subblock", but rather whether "an operation of **permuting subkey bits** is performed **on the subkey** depending on the value of a j-th data subblock (emphasis added," as required by claim 1. As discussed above, since the "P-Box Permutation" takes the XOR combination of the subkey and the data subblock, the permutation performed by the "P-Box Permutation" effectively permutates the subkey in a manner dependant upon the value of the data subblock since they are effectively combined via an XOR operation. The manner of how this input is permuted is not relevant to the claims.

Appellant goes on to describe how the subkeys are formed and specifying that the keys are not formed based on the data subblock being converted. Examiner wishes to reiterate the point that this aspect of the DES algorithm described by Schneier was not relied upon in the rejections of claims 1, 3, and 5. Rather the "P-Box Permutation" of the subkey and data subblock were relied upon to rejection the claims (as discussed in detail above).

Appellant goes on to allege that "the conversion operation *f* is not a permutation operation." This allegation is not persuasive because Schneier describes the "P-Box Permutation" as a "straight permutation" (see page 275, line 3 of paragraph under **The P-Box Permutation** section, included herewith).

In the paragraph beginning at the end of page 5 of the Appeal Brief, which carries over to page 6, Appellant continues to allege that the *f* operation of Schneier does not include permutation. However, Schneier in fact describes at least the "P-Box Permutation" as a "straight permutation" (see page 275, line 3 of paragraph under **The P-Box Permutation** section). The claims do not include any claim limitations that effectively differentiate the claimed "permuting" from the "straight permutation" as described by Schneier.

Appellant argues, "Schneier does not explain nor describe that the conversion operation *f* is a permutation operation." This argument is not persuasive because Schneier discloses that the "P-Box Permutation", which is a part of the conversion operation *f* (as discussed by Schneier on Page 270, second paragraph under **Outline of the Algorithm**), is a "straight permutation" (see page 275, line 3 of paragraph under **The P-Box Permutation** section).

Appellant continues to allege that the *f* operation of Schneier does not include permutation (last paragraph on page 6 of the Appeal Brief, which carries over to page 7). However, the claims do not include any claim limitations that effectively differentiate the claimed "permuting" from the "straight permutation" as described by Schneier.

Appellant aruges, "the conversion operation *f* is not a bit permutation operation and does not include any bit permutation operation dependent on a data subblock." This argument is not persuasiave because Schneier disclsoes on page 275, first paragraph under **The P-Box Permutation** section:

> This permutation maps each input bit to an output position; no bits are used twice and no bits are ignored. This is called a **straight permutation** or just a permutation.

Therefore, it is clear that the "P-Box Permutation" is a bit permutation operation. Examiner has fully addressed how Schneier meets "permutation operation dependent on a data subblock" in the remarks above.

Appellant argues, "although page 270 of Schneier indicates that 'The right half of the data is expanded to 48 bits via an expansion permutation, combined with 48 bits of a shifted and permuted key via an Xor…', one of ordinary skill in the art cannot conclude from this reference that a bit permutation operation was previously performed on the subkey **depending on** some **data subblock** being converted." Again, the Examiner is relying on "a bit permutation operation…previously performed" to meet the claimed "permuting", but is instead relying on the "P-Box Permutation" to read on the claimed "permuting", which has been fully addressed above.

Appellant argues, "Thus, in algorithm DES, the **bit permutation operation** is performed on the key by **depending on the number of the round, but not on the data subblock, i.e. algorithm DES lacks the feature performing the subkey bit permutation operation**

depending on the data subblock being converted, that is presented in the claimed invention." This argument is not persuasive because Examiner is not relying on the "Key Transformation" to read on the claimed "permuting", but is instead relying on the "P-Box Permutation" to read on the claimed "permuting", which has been fully addressed above.

Appellant's final commentary continues to misinterpret the Examiner's reliance on the teachings of Schneier to meet the claimed "permuting." Examiner believes his position on how the "P-Box Permutation" of Schneier meets the claimed "permuting" to have been fully addressed above.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Benjamin E. Lanier
Examiner 2132
Granted Temporary Full Signatory Authority

Conferees:

Gilberto Barron GB
SPE 2132

Matthew Smithers

/Matthew Smithers/
Primary Examiner, AU 2137

Table 12.5
Expansion Permutation

32,	1,	2,	3,	4,	5,	4,	5,	6,	7,	8,	9,
8,	9,	10,	11,	12,	13,	12,	13,	14,	15,	16,	17,
16,	17,	18,	19,	20,	21,	20,	21,	22,	23,	24,	25,
24,	25,	26,	27,	28,	29,	28,	29,	30,	31,	32,	1

responds to row 3 of the sixth S-box. The middle 4 bits combine to form 1001, which corresponds to the column 9 of the same S-box. The entry under row 3, column 9 of S-box 6 is 14. (Remember to count rows and columns from 0 and not from 1.) The value 1110 is substituted for 110011.

It is, of course, far easier to implement the S-boxes in software as 64-entry arrays. It takes some rearranging of the entries to do this, but that's not hard. (Don't just change the indexing without rearranging the entries. The S-boxes are designed very carefully.) However, this way of describing the S-boxes helps visualize how they work. Each S-box can be viewed as a substitution function on a 4-bit entry: b_2 through b_5 go in, and a 4-bit result comes out. Bits b_1 and b_6 come from neighboring blocks; they select one out of four substitution functions available in the particular S-box.

The S-box substitution is the critical step in DES. The algorithm's other operations are linear and easy to analyze. The S-boxes are nonlinear and, more than anything else, give DES its security.

The result of this substitution phase is eight 4-bit blocks which are recombined into a single 32-bit block. This block moves to the next step: the P-box permutation.

The P-Box Permutation

The 32-bit output of the S-box substitution is permuted according to a **P-box**. This permutation maps each input bit to an output position; no bits are used twice and no bits are ignored. This is called a **straight permutation** or just a permutation. Table 12.7 shows the position to which each bit moves. For example, bit 21 moves to bit 4, while bit 4 moves to bit 31.

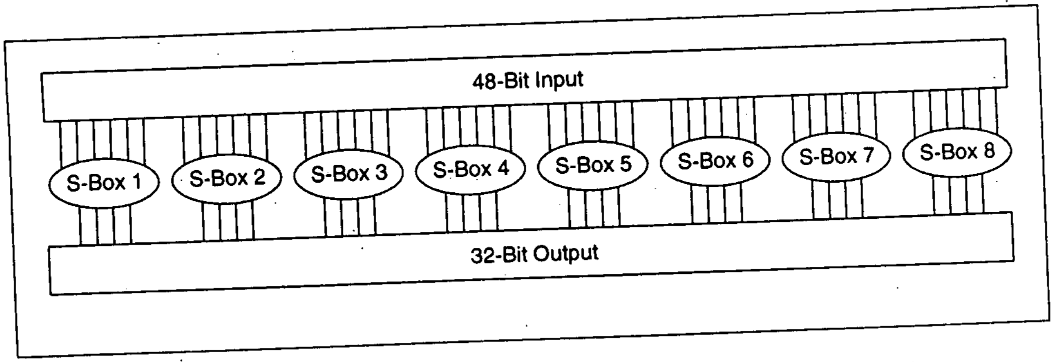

Figure 12.4 S-box substitution.